**Udacity - Artificial Intelligence Nanodegree Program**

**Project: Forward Planning Agent**
**Student: Jair Miranda França**
**Submission Date: July, 25, 2021**

## 1 Introduction:

The project is to build a planning graphic with various agents to solve a deterministics logistics planning problem; Air Cargo Transport System.
I build a planning agent to efficiently plan a strategy to reach the final goal.

We have 4 different cargo problems, and we'll describe below several different search algorithms.

### 1.1 Cargo Problems:

1. Cargos; C1, C2, Planes; P1, P2, Airports; JFK, SFO.
2. Cargos: C1, C2, C3, Planes; P1, P2, P3 Airports; JFK, SFO, ATL.
3. Cargos: C1, C2, C3, C4, Planes; P1, P2, Airports; JFK, SFO, ATL, ORD.
4. Cargos: C1, C2, C3, C4, C5 Planes; P1, P2, Airports; JFK, SFO, ATL, ORD.

## 1.2 Functions Implemented:

**1.2.1 Action Layer -** Define mutex relation between two action in a given level:

1. _incosistent_effects: The Idea is to check if any effect in actionA is a negation of actionB.
2. _inference: Check if any effect of action is negation of a precondition of the other.
3. _competing_need: Check if the precondition of one level is mutually exclusive with a precondition of a parent layer.

**1.2.2 Literal Layer -** Layer of states, logical condition.

1. _inconsistent_support: Check if actions in a layer and parent layer are pairwise mutex.
2. _negation: Check one literal is a negation of another.

**1.2.3 Heuristics on a Planning Graph -** Using the knowledge about the problem to find the solution in a more efficient way.

1. h_levelsum - Calculate the sum of levels of all goals until we find the target.
2. h_maxlevel - Calculate the max level cost of all layers to reach the target.
3. h_setlevel - Find the first level layer where all goals are not mutex .

## 2- Results:

### 2.1 Reports on Each problem.

| Problem 1 - | | | | | |
|---|---|---|---|---|---|
| **Actions** | **20** | | | | |
| **Search Algorithm** | **Expansion** | **Goal Tests** | **New Nodes** | **Time (s)** | **Plan Length** |
| Breadth First Search | 43 | 56 | 178 | 0,00485 | 6 |
| Depth First Graph Search | 21 | 22 | 84 | 0,00257 | 20 |
| Uniform Cost Search | 60 | 62 | 240 | 0,0072 | 6 |
| **Greedy Best First Graph Search** | | | | | |
| Heuristic unmet Goals | 7 | 9 | 29 | 0,00122 | 6 |
| Heuristic Planning Graphic Sum | 6 | 8 | 28 | 0,09187 | 6 |
| Heuristic Planning Graphic Max Level | 6 | 8 | 24 | 0,06683 | 6 |
| Heuristic Planning Graphic Set Level | 6 | 8 | 28 | 0,12835 | 6 |
| **A* Graph Search** | | | | | |
| Heuristic unmet Goals | 50 | 52 | 206 | 0,00354 | 6 |
| Heuristic Planning Graphic Sum | 28 | 30 | 122 | 0,22651 | 6 |
| Heuristic Planning Graphic Max Level | 43 | 45 | 180 | 0,25588 | 6 |
| Heuristic Planning Graphic Set Level | 33 | 35 | 138 | 0,33378 | 6 |

| Problem 2 - | | | | | |
|---|---|---|---|---|---|
| **Actions** | **72** | | | | |
| **Search Algorithm** | **Expansion** | **Goal Tests** | **New Nodes** | **Time (s)** | **Plan Length** |
| Breadth First Search | 3343 | 4609 | 30503 | 0,73592 | 9 |
| Depth First Graph Search | 642 | 625 | 5602 | 0,99041 | 619 |
| Uniform Cost Search | 5154 | 5156 | 46618 | 1,49222 | 9 |
| Greedy Best First Graph Search | | | | | |
| Heuristic unmet Goals | 17 | 19 | 170 | 0,00911 | 9 |
| Heuristic Planning Graphic Sum | 9 | 11 | 86 | 2,16440 | 9 |
| Heuristic Planning Graphic Max Level | 27 | 29 | 249 | 4,53520 | 9 |
| Heuristic Planning Graphic Set Level | 9 | 11 | 84 | 3,41640 | 9 |
| A* Graph Search | | | | | |
| Heuristic unmet Goals | 2467 | 2469 | 22522 | 0,88160 | 9 |
| HeuristicPlanning Graphic Sum | 357 | 359 | 3426 | 58,16600 | 9 |
| Heuristic Planning Graphic Max Level | 2887 | 2889 | 26594 | 347,19510 | 9 |
| Heuristic Planning Graphic Set Level | 1037 | 1039 | 9605 | 318,19710 | 9 |

| Problem 3 - | | | | | |
|---|---|---|---|---|---|
| **Actions** | **88** | | | | |
| **Search Algorithm** | **Expansion** | **Goal Tests** | **New Nodes** | **Time (s)** | **Plan Length** |
| Breadth First Search | 14663 | 18098 | 129625 | 3,86499 | 12 |
| Greedy Best First Graph Search | | | | | |
| Heuristic unmet Goals | 25 | 27 | 230 | 0,0133 | 15 |
| Heuristic Planning Graphic Sum | 14 | 16 | 126 | 4,8675 | 14 |
| A* Graph Search | | | | | |
| Heuristic unmet Goals | 7388 | 7390 | 65711 | 3,28572 | 12 |
| Heuristic Planning Graphic Sum | 369 | 371 | 3403 | 86,67390 | 12 |

| Problem 4 - | | | | | |
|---|---|---|---|---|---|
| **Actions** | **104** | | | | |
| **Search Algorithm** | **Expansion** | **Goal Tests** | **New Nodes** | **Time (s)** | **Plan Length** |
| Breadth First Search | 99736 | 114953 | 944130 | 34,33390 | 14 |
| Greedy Best First Graph Search | | | | | |
| Heuristic unmet Goals | 29 | 31 | 280 | 0,02409 | 18 |
| Heuristic Planning Graphic Sum | 17 | 19 | 165 | 8,23347 | 17 |
| A* Graph Search | | | | | |
| Heuristic unmet Goals | 34330 | 34332 | 328509 | 20,726256 | 14 |
| Heuristic Planning Graphic Sum | 1208 | 1210 | 12210 | 483,59964 | 15 |

**2.2 The worst algorithm**

On the two first problems we can see the deep first graph search algorithm is unique to find the most plan length bigger than 6, so we'll not use it on problems 3 and 4.

**2.3 Time to find solution analyses**

Comparing problem one and two we have a significant increase in time to find solutions, especially those using A* search, graphic sum , max level and set level. Of course we have more than 3 times of action on problem 2.

So to run problem 3 and 4 we chose the more efficient algorithms:

1. Uninformed search; Breadth First Search.
2. Greedy Best First Graph Search: Heuristic Unmet Goals, Heuristic Planning Graphic Sum.
3. A* Graph Search: Heuristic Unmet Goals, Heuristic Planning Graphic Sum.

On problem 3, more complex than one and two, we can saw different behavior, 3 of 5 combined algorithms reach the best plan Length, BFS, A* unmet goals and A* with sum of levels, none of options with GBF reach the best plan length, and the fastest option is the A* unmet goals.

On problem 4, the most complex on all 4, only 2 of 5 reached the optimal plan length, the BFS and A* with Unmet goals, and the fastest option was is the A* with Unmet goals.

**2.4 Relation between actions and expanded nodes.**

As we can see when the actions increase the expansion increase too, and of corse new nodes are generated

## 3- Questions to answer.

### 3.1 Which algorithm or algorithms would be most appropriate for planning in a very restricted domain (i.e., one that has only a few actions) and needs to operate in real time?

GBF, unmet goal, A* unmet goal and BFS, this 3 can reach the best plan length and and very fast for a streaming application.

### 3.2 Which algorithm or algorithms would be most appropriate for planning in very large domains (e.g., planning delivery routes for all UPS drivers in the U.S. on a given day)?

For the large domain i should use A* with unmet goals, was the fastest and reached the optimal plan length.

### 3.2 Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?

The only two algorithms that find the optimal plans was the A* with unmet Goals and the BFS.